

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

1. Creating Jenkins Puppet Class

```
class jenkins {
```

```
[root@master manifests]# cat init.pp
#Puppet code to install jenkins on CentOS 9
class jenkins {
# -----
# Jenkins configuration
```

What this does

- Defines a Puppet class named jenkins

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

2. Configuring Jenkins Repository

```
# Jenkins repository
# -----
yumrepo { 'jenkins':
  baseurl => 'https://pkg.jenkins.io/redhat-stable',
  descr => 'Jenkins Stable Repo',
  enabled => 1,
  gpgcheck => 1,
  gpgkey => 'https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key',
}
```

What this does

- Adds the official Jenkins stable repository
- Enables the repository on the system
- Configures GPG key verification for security

```
[root@master manifests]# cat init.pp
#Puppet code to install jenkins on CentOS 9
class jenkins {
  # -----
  # Jenkins repository
  # -----
  yumrepo { 'jenkins':
    baseurl => 'https://pkg.jenkins.io/redhat-stable',
    descr => 'Jenkins Stable Repo',
    enabled => 1,
    gpgcheck => 1,
    gpgkey => 'https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key',
  }
}
```

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

3. Configuring Adoptium (Java) Repository

```
# Adoptium repository
# -----
yumrepo { 'adoptium':
  baseurl =>
  'https://packages.adoptium.net/artifactory/rpm/rhel/$releasever/$basearch',
  descr => 'Adoptium',
  enabled => 1,
  gpgcheck => 1,
  gpgkey => 'https://packages.adoptium.net/artifactory/api/gpg/key/public',
}
```

```
# Adoptium repository
# -----
yumrepo { 'adoptium':
  baseurl => 'https://packages.adoptium.net/artifactory/rpm/rhel/$releasever/$basearch',
  descr => 'Adoptium',
  enabled => 1,
  gpgcheck => 1,
  gpgkey => 'https://packages.adoptium.net/artifactory/api/gpg/key/public',
}
```

What this does

- Adds Adoptium repository for OpenJDK
- Allows installation of Java 21 on CentOS 9

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

4. Installing Required Packages

```
# Required packages
# -----
package { ['fontconfig', 'java-21-openjdk']:
  ensure => present,
}
package { 'jenkins':
  ensure => present,
  require => Yumrepo['jenkins'],
}
```

```
# -----
# Required packages
# -----
package { ['fontconfig', 'java-21-openjdk']:
  ensure => present,
}
package { 'jenkins':
  ensure => present,
  require => Yumrepo['jenkins'],
}
```

What this does

- Installs Java 21 OpenJDK
- Installs fontconfig dependency. ontconfig is required for Jenkins UI rendering
- Puppet ensures packages are installed only if missing
- Installs Jenkins package
- Ensures Jenkins repository is configured before installation

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

5. Creating systemd Override Directory

```
# systemd override directory
# -----
file { '/etc/systemd/system/jenkins.service.d':
  ensure => directory,
  mode  => '0755',
}
```

```
# systemd override directory
# -----
file { '/etc/systemd/system/jenkins.service.d':
  ensure => directory,
  mode  => '0755',
}
```

What this does

- Creates a systemd override directory for Jenkins service

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

6. Configuring Jenkins Port

```
# Jenkins port configuration
# -----
file { '/etc/systemd/system/jenkins.service.d/override.conf':
  ensure => file,
  mode => '0644',
  content => "[Service]\nEnvironment=\"JENKINS_PORT=8000\"\n",
  notify => Exec['systemd-daemon-reload'],
}
```

```
# Jenkins port configuration
# -----
file { '/etc/systemd/system/jenkins.service.d/override.conf':
  ensure => file,
  mode => '0644',
  content => "[Service]\nEnvironment=\"JENKINS_PORT=8000\"\n",
  notify => Exec['systemd-daemon-reload'],
}
```

What this does

- Configures Jenkins to run on port 8000
- Triggers systemd reload when configuration changes

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

7. Reloading systemd Configuration

```
# Reload systemd
# -----
exec { 'systemd-daemon-reload':
  command
=> '/bin/systemctl daemon-reload',
  refreshonly => true,
}
```

```
"# Reload systemd
# -----
exec { 'systemd-daemon-reload':
  command
=> '/bin/systemctl daemon-reload',
  refreshonly => true,
}
```

What this does

- Reloads systemd configuration only when required

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

8. Managing Jenkins Service

```
# Jenkins service
# -----
service { 'jenkins':
  ensure => running,
  enable => true,
}
```

```
# Jenkins service
# -----
service { 'jenkins':
  ensure => running,
  enable => true,
}
```

What this does

- Starts Jenkins service
- Enables Jenkins to start on system boot

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

9. Opening Firewall Port

```
# Firewall rule
# -----
exec { 'open-jenkins-port':command => '/usr/bin/firewall-cmd --permanent
--add-port=8000/tcp',
unless => '/usr/bin/firewall-cmd --list-ports | grep -q 8000/tcp',
notify => Exec['firewalld-reload'],
}
```

```
# Firewall rule
# -----
exec { 'open-jenkins-port':command => '/usr/bin/firewall-cmd --permanent --add-port=8000/tcp',
unless => '/usr/bin/firewall-cmd --list-ports | grep -q 8000/tcp',
notify => Exec['firewalld-reload'],
}
exec { 'firewalld-reload':
```

What this does

- Opens TCP port 8000 in firewall
- Prevents duplicate firewall rules

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

10. Reloading Firewall Rules

```
exec { 'firewalld-reload':  
  command  
  => '/usr/bin/firewall-cmd --reload',  
  refreshonly => true,  
}
```

```
exec { 'firewalld-reload':  
  command  
  => '/usr/bin/firewall-cmd --reload',  
  refreshonly => true,  
}
```

What this does

- Reloads firewall rules only when required

Puppet Code Explanation: Jenkins Installation and Configuration on CentOS 9 Using Puppet

11. Closing the Class

```
}
```

```
exec { 'firewalld-reload':  
    command  
    => '/usr/bin/firewall-cmd --reload',  
    refreshonly => true,  
}  
}
```

What this does

- Ends the Jenkins class definition